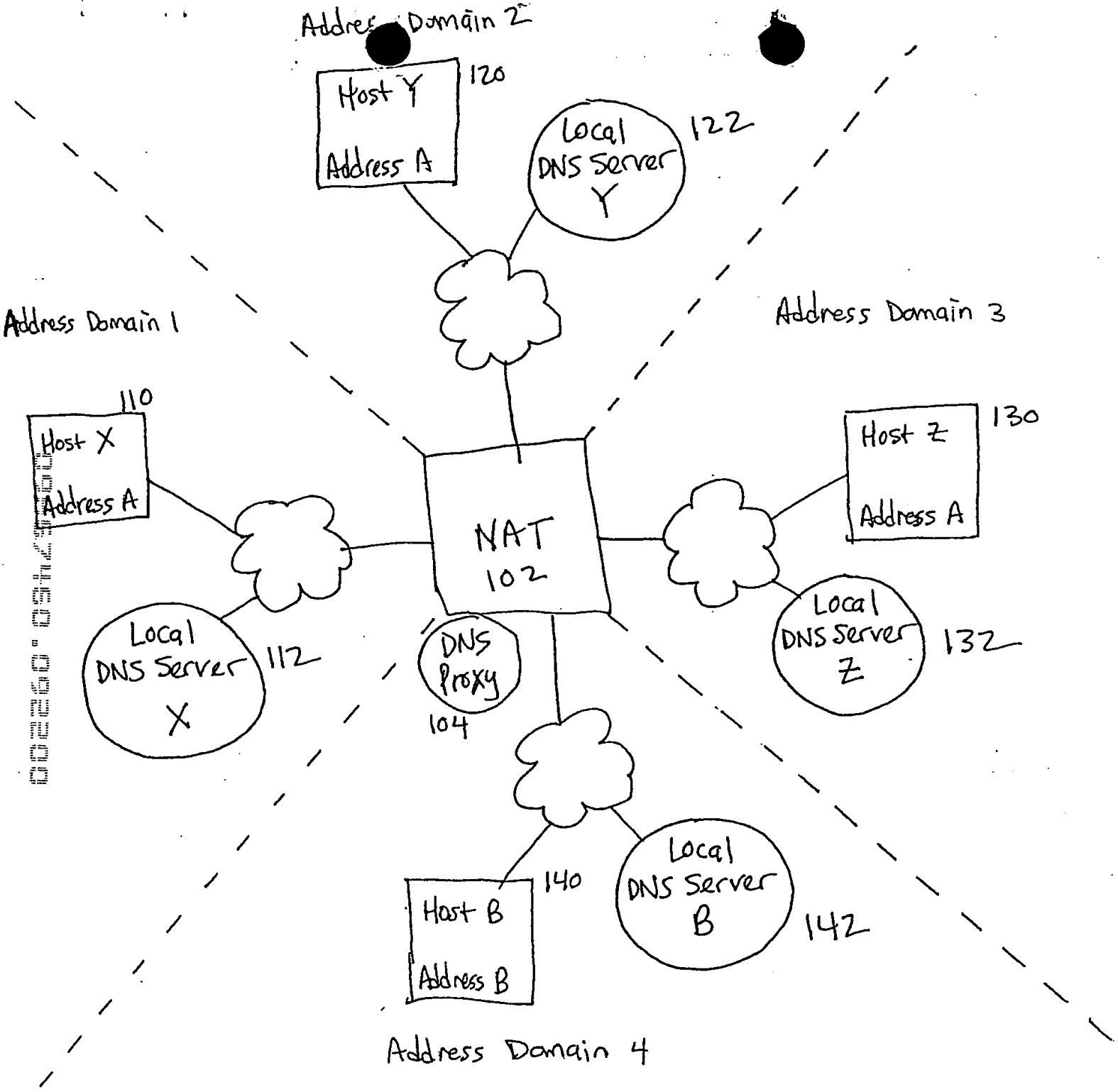
Address Domain 2
Host Y
Address A
120
Local
DNS Server
Y
122
Address Domain 1
Address Domain 3
110
Host X
Address A
Host Z
Address A
130
NAT
102
Local
DNS Server
X
112
DNS
Proxy
104
Local
DNS Server
Z
132
Host B
Address B
140
Local
DNS Server
B
142
Address Domain 4

FIG. 1 100

	Source Local Address	Protocol	Source Port	Source Address Domain	Destination Address Domain	Translated Source Port	Source Global Address
202	A (host X)			1	2		A12
204	A (host X)			1	3		A13
206	A (host X)			1	4		A14

FIG. 2A

	Source Local Address	Protocol	Source Port	Source Address Domain	Destination Address Domain	Translated Source Port	Source Global Address
208	A (host Y)			2	1		A21
210	A (host Y)			2	3		A23
212	A (host Y)			2	4		A24

FIG. 2B

	Source Local Address	Protocol	Source Port	Source Address Domain	Destination Address Domain	Translated Source Port	Source Global Address
214	A (host Z)			3	1		A31
216	A (host Z)			3	2		A32
218	A (host Z)			3	4		A34

FIG. 2C

	Destination Global Address	Protocol	Translated Destination Port	Source Address Domain	Destination Address Domain	Destination Port	Destination Local Address
220	A12			2	1		A (host X)
222	A13			3	1		A (host X)
224	A14			4	1		A (host X)
226	A21			1	2		A (host Y)
228	A23			3	2		A (host Y)
230	A24			4	2		A (host Y)
232	A31			1	3		A (host Z)
234	A32			2	3		A (host Z)
236	A34			4	3		A (host Z)
238	B			0	4		B

FIG. 2D

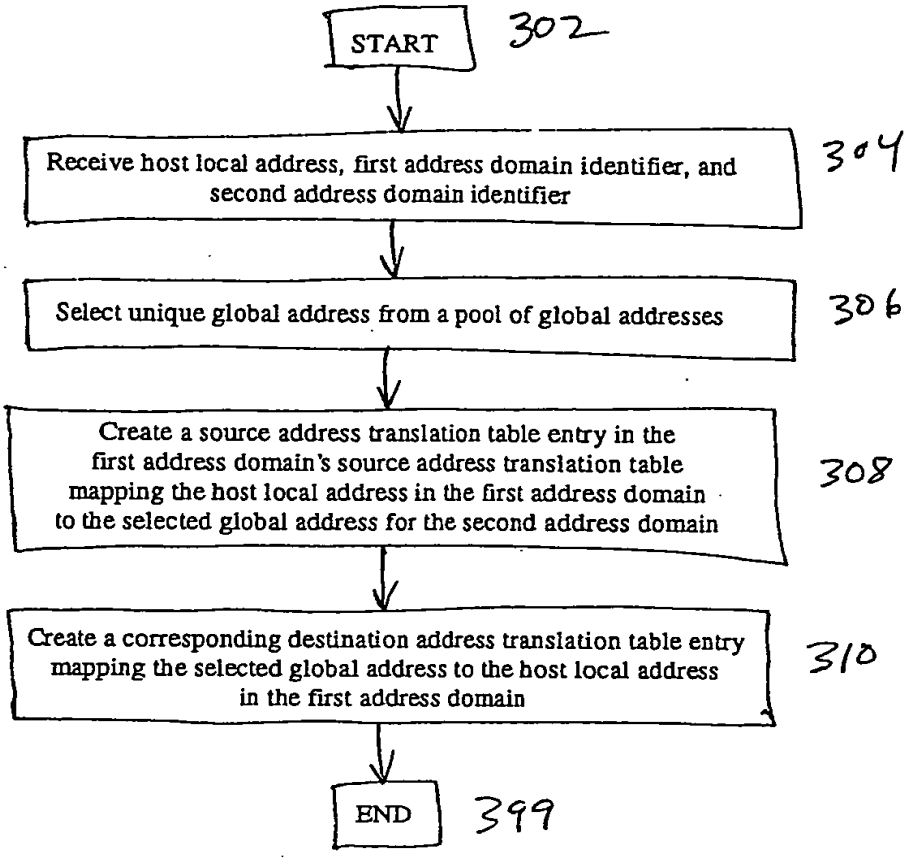
START
302
Receive host local address, first address domain identifier, and second address domain identifier
304
Select unique global address from a pool of global addresses
306
Create a source address translation table entry in the first address domain's source address translation table mapping the host local address in the first address domain to the selected global address for the second address domain
308
Create a corresponding destination address translation table entry mapping the selected global address to the host local address in the first address domain
310
END
399

FIG. 3

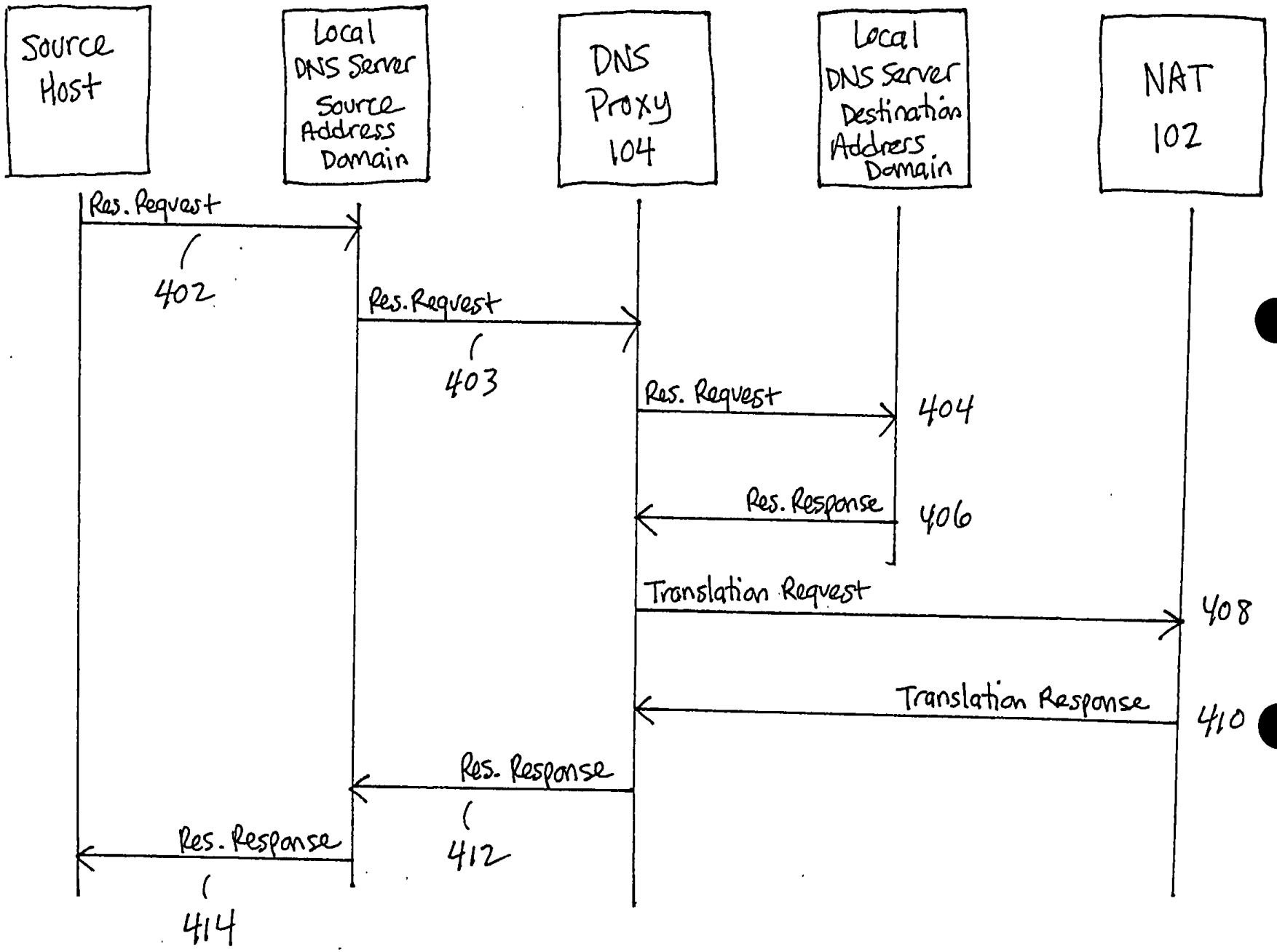

FIG. 4

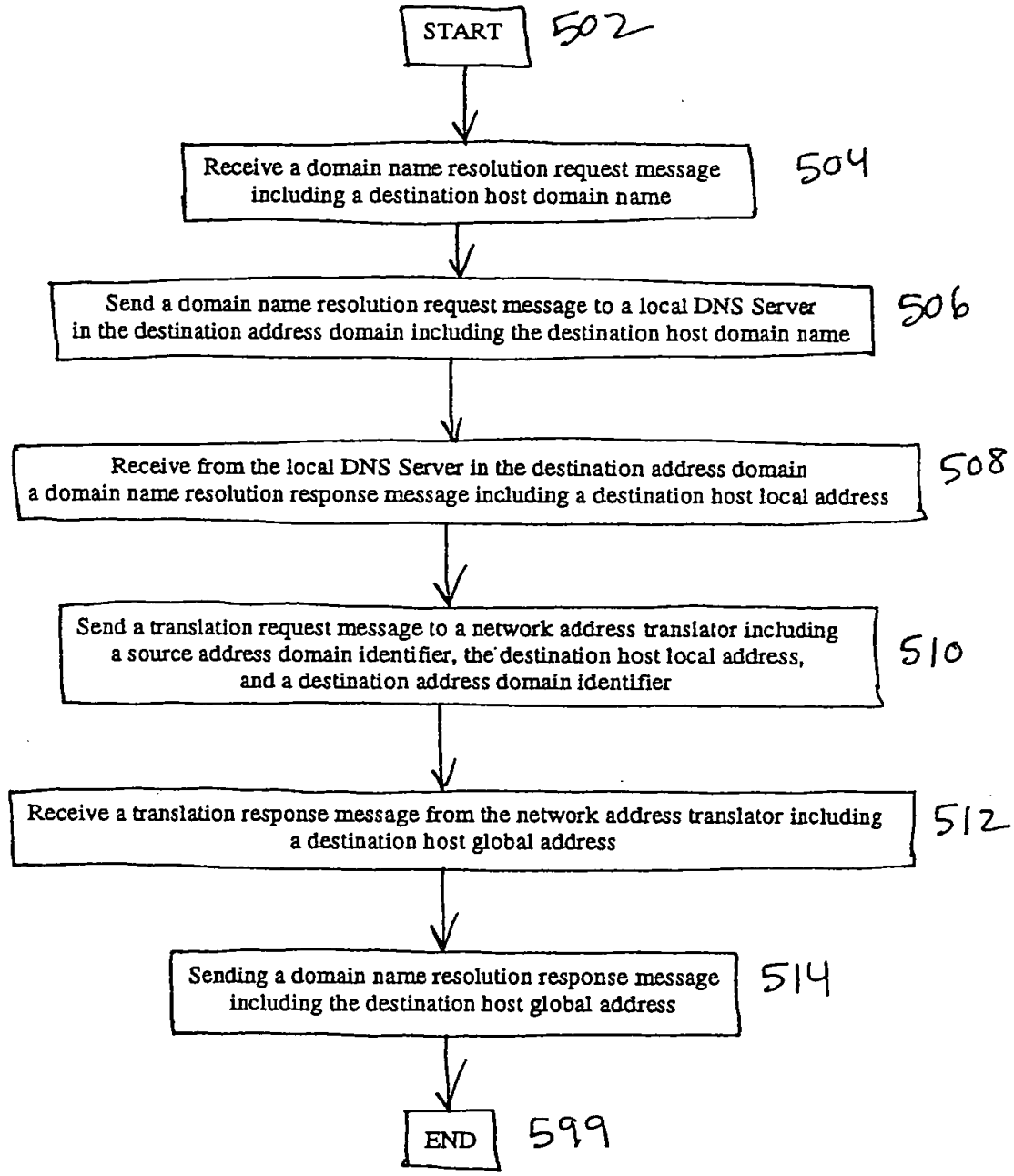
START
502
Receive a domain name resolution request message including a destination host domain name
504
Send a domain name resolution request message to a local DNS Server in the destination address domain including the destination host domain name
506
Receive from the local DNS Server in the destination address domain a domain name resolution response message including a destination host local address
508
Send a translation request message to a network address translator including a source address domain identifier, the destination host local address, and a destination address domain identifier
510
Receive a translation response message from the network address translator including a destination host global address
512
Sending a domain name resolution response message including the destination host global address
514
END
599

FIG. 5

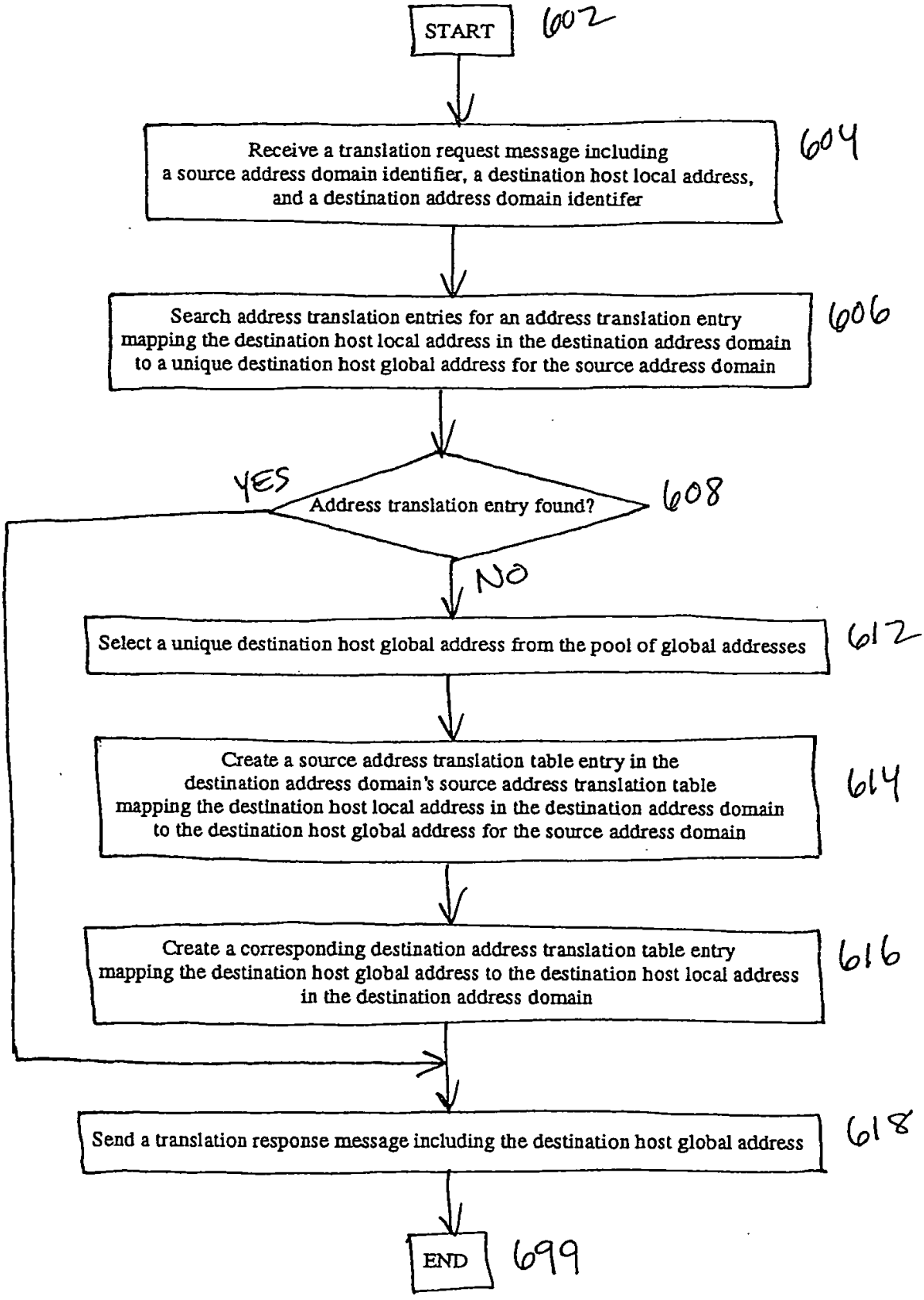
START
602
Receive a translation request message including a source address domain identifier, a destination host local address, and a destination address domain identifer
604
Search address translation entries for an address translation entry mapping the destination host local address in the destination address domain to a unique destination host global address for the source address domain
606
YES
Address translation entry found?
608
NO
Select a unique destination host global address from the pool of global addresses
612
Create a source address translation table entry in the destination address domain's source address translation table mapping the destination host local address in the destination address domain to the destination host global address for the source address domain
614
Create a corresponding destination address translation table entry mapping the destination host global address to the destination host local address in the destination address domain
616
Send a translation response message including the destination host global address
618
END
699

FIG. 6

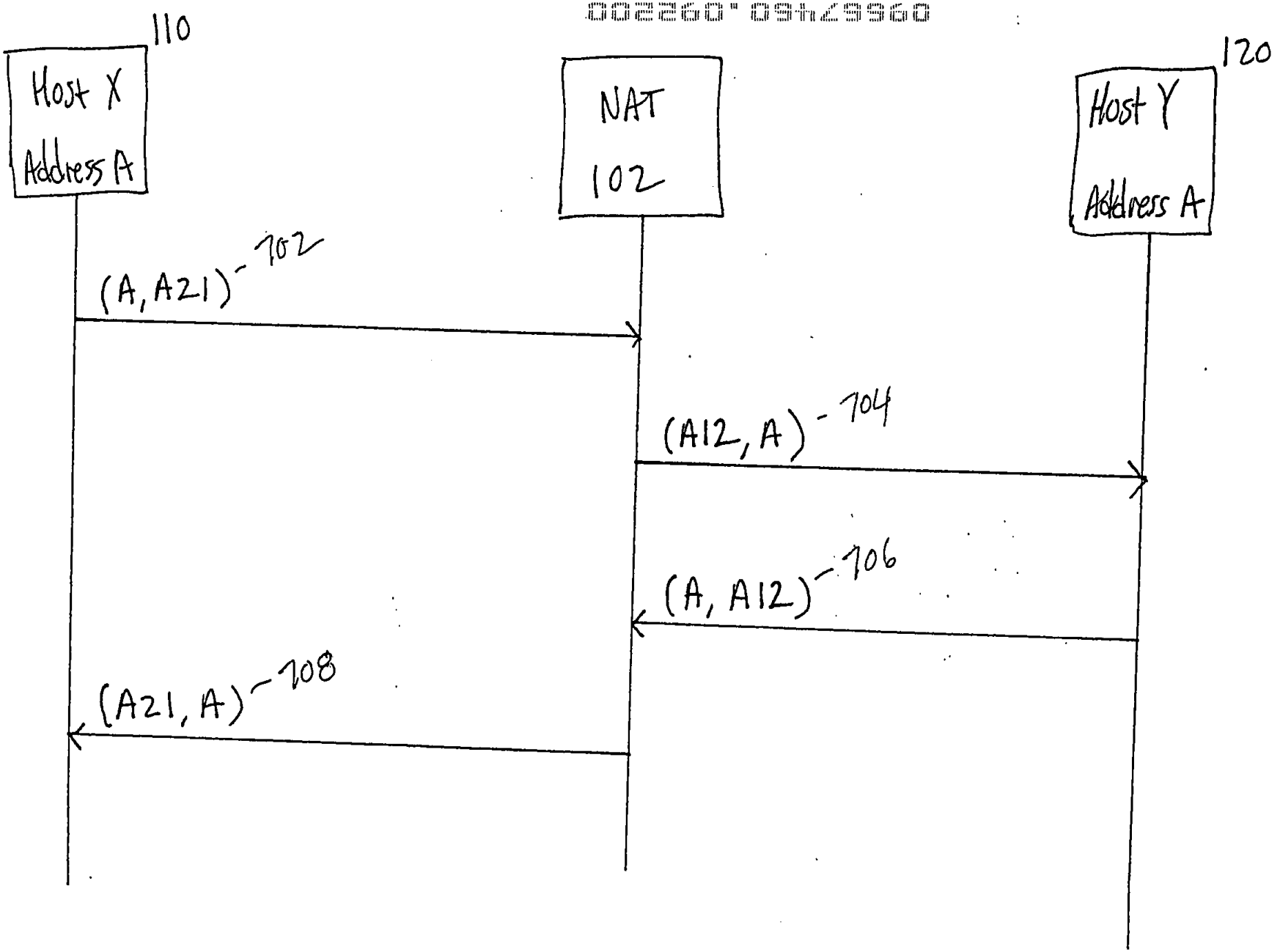
110
Host X
Address A
NAT
102
120
Host Y
Address A
(A, A21) - 702
(A12, A) - 704
(A, A12) - 706
(A21, A) - 708

FIG. 7

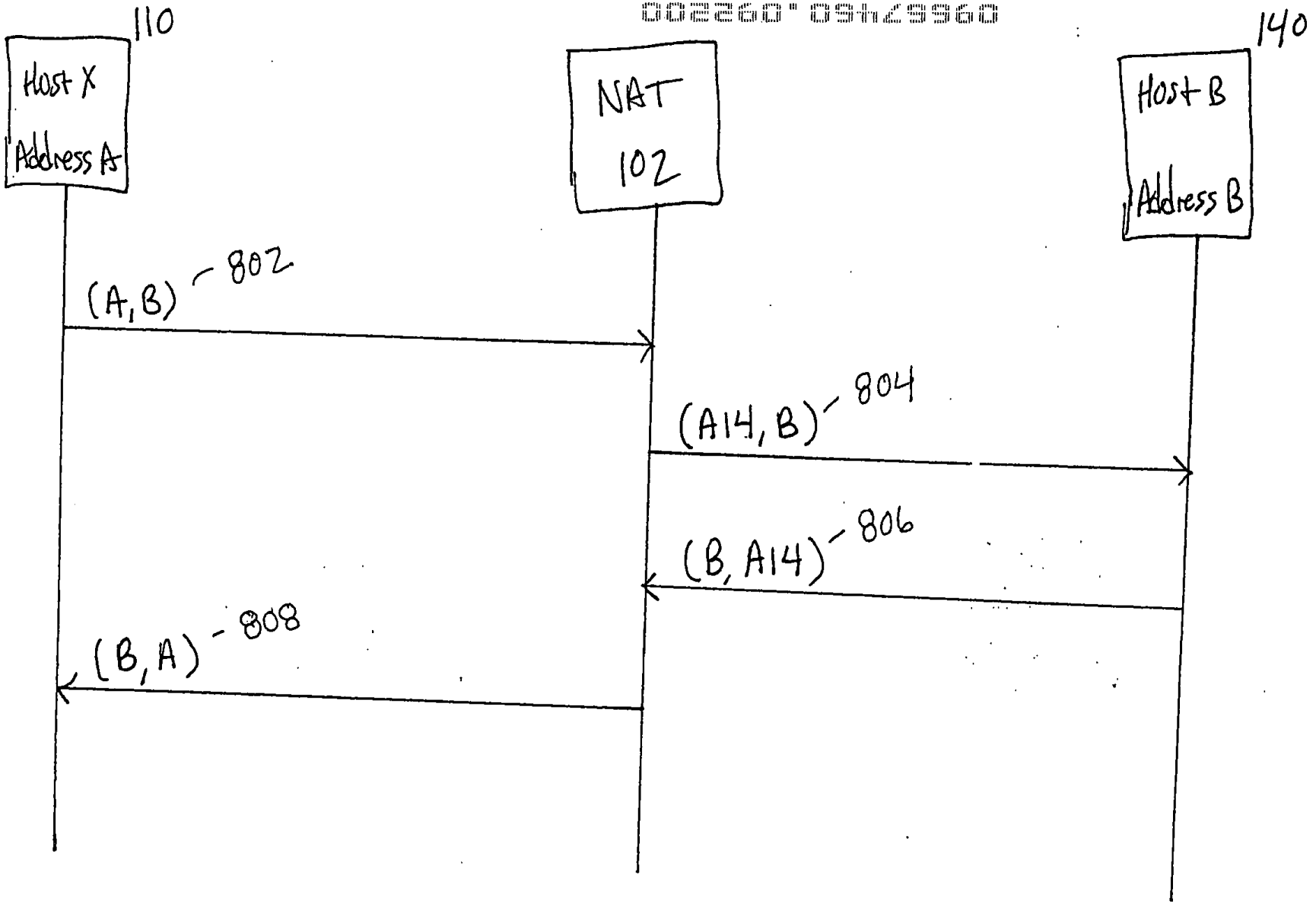

FIG. 8

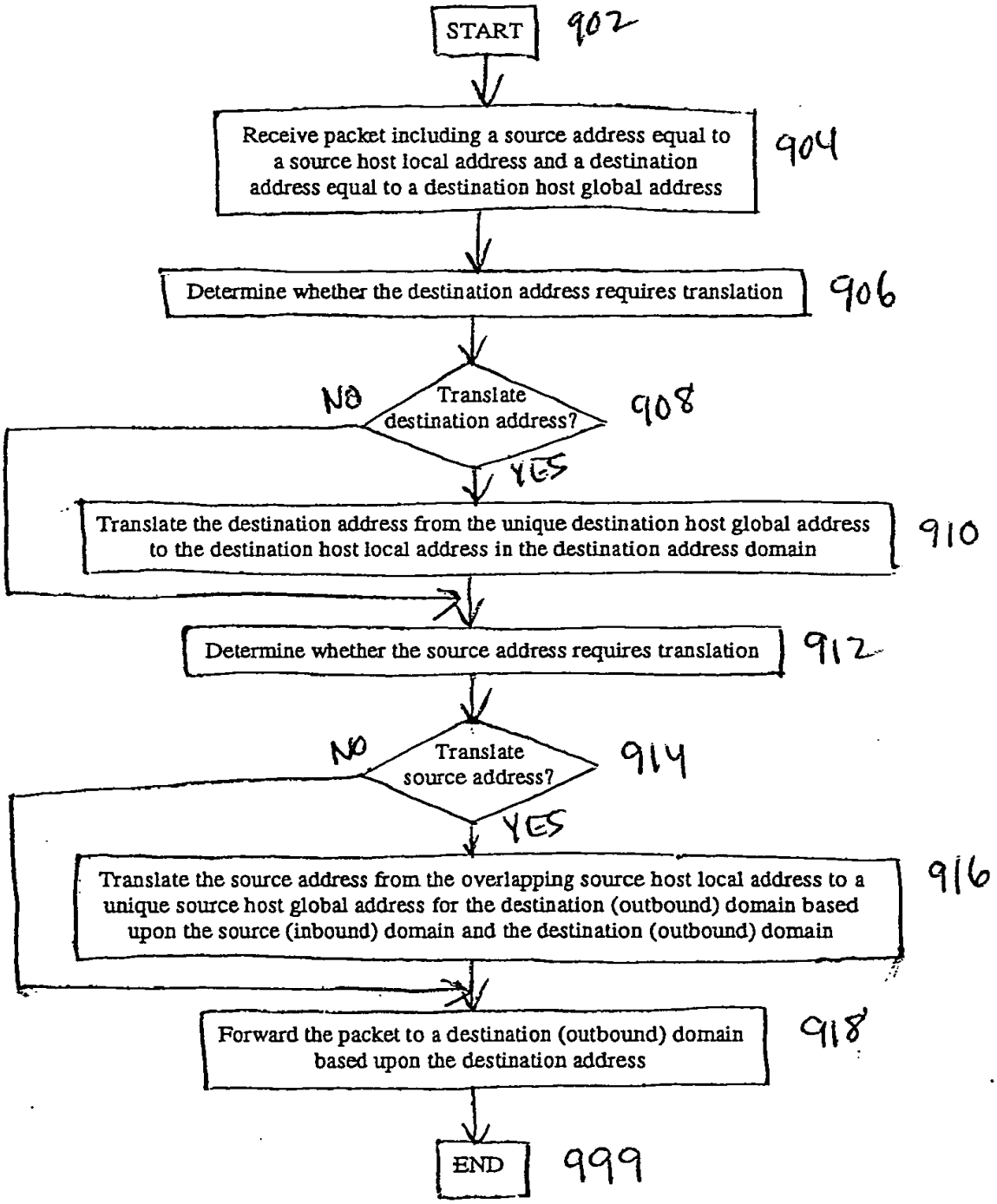
START
902
Receive packet including a source address equal to a source host local address and a destination address equal to a destination host global address
904
Determine whether the destination address requires translation
906
NO
Translate destination address?
908
YES
Translate the destination address from the unique destination host global address to the destination host local address in the destination address domain
910
Determine whether the source address requires translation
912
NO
Translate source address?
914
YES
Translate the source address from the overlapping source host local address to a unique source host global address for the destination (outbound) domain based upon the source (inbound) domain and the destination (outbound) domain
916
Forward the packet to a destination (outbound) domain based upon the destination address
918
END
999

FIG. 9

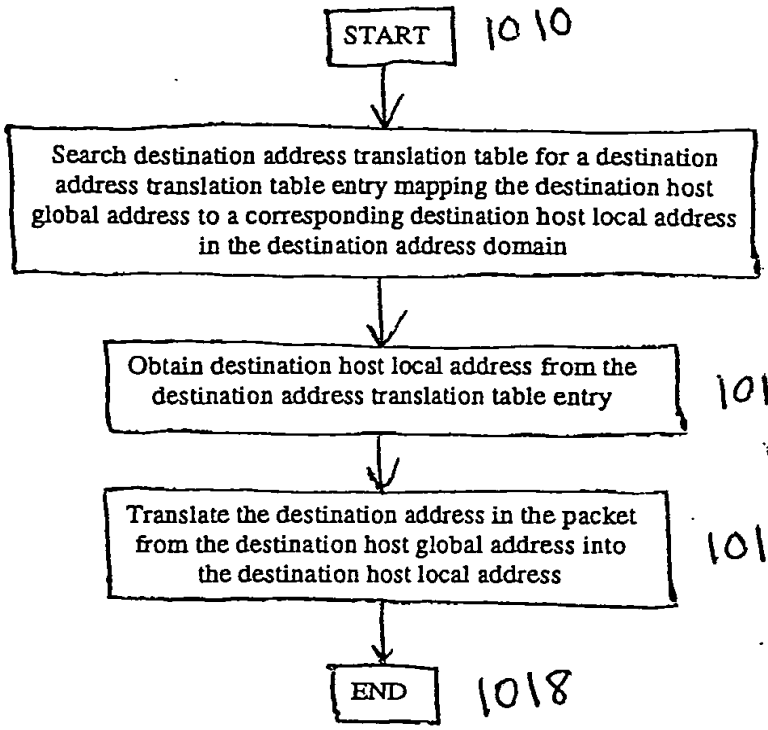

FIG. 10A 910

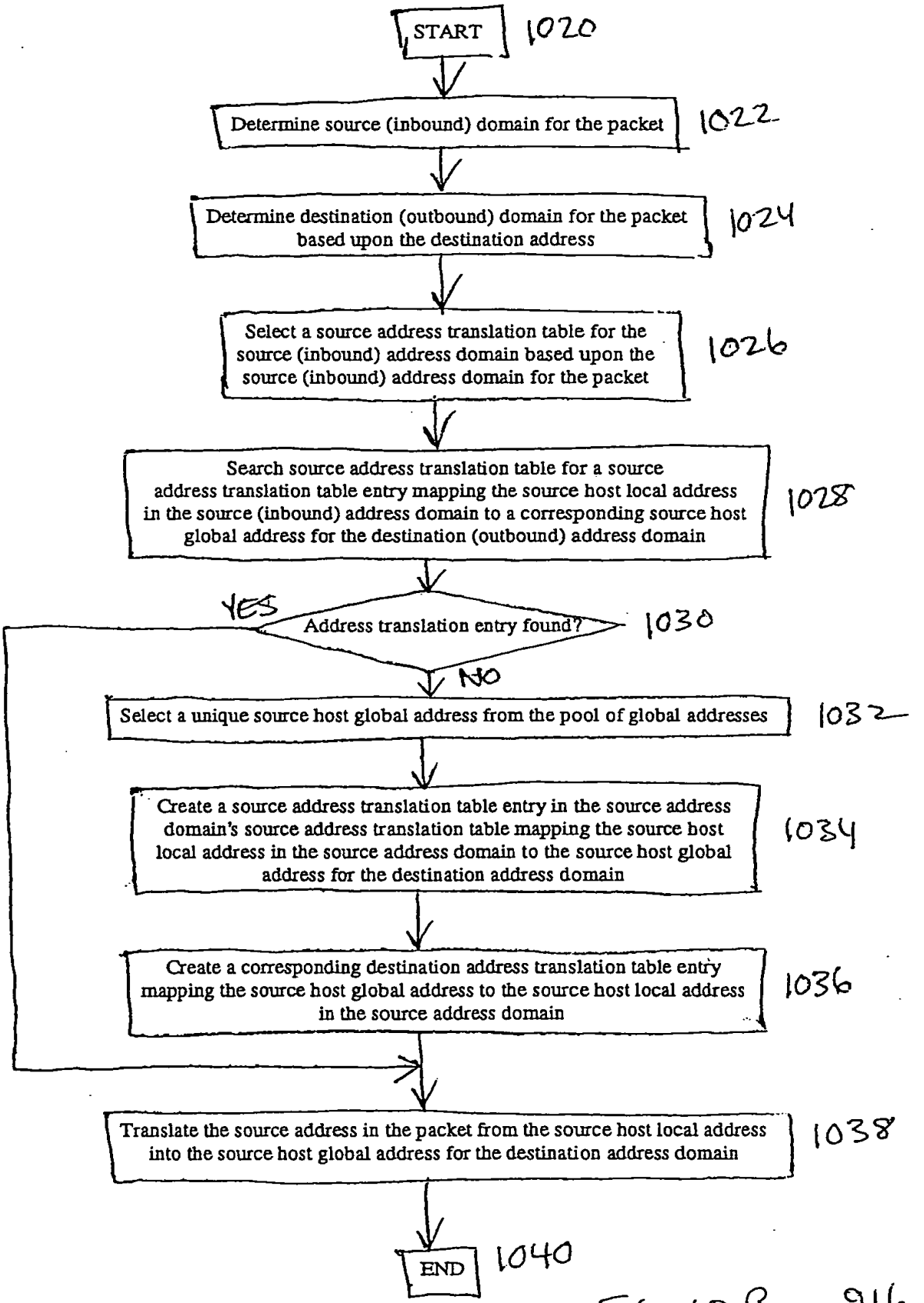
START
1020
Determine source (inbound) domain for the packet
1022
Determine destination (outbound) domain for the packet based upon the destination address
1024
Select a source address translation table for the source (inbound) address domain based upon the source (inbound) address domain for the packet
1026
Search source address translation table for a source address translation table entry mapping the source host local address in the source (inbound) address domain to a corresponding source host global address for the destination (outbound) address domain
1028
YES
Address translation entry found?
1030
NO
Select a unique source host global address from the pool of global addresses
1032
Create a source address translation table entry in the source address domain's source address translation table mapping the source host local address in the source address domain to the source host global address for the destination address domain
1034
Create a corresponding destination address translation table entry mapping the source host global address to the source host local address in the source address domain
1036
Translate the source address in the packet from the source host local address into the source host global address for the destination address domain
1038
END
1040

FIG. 10B 916

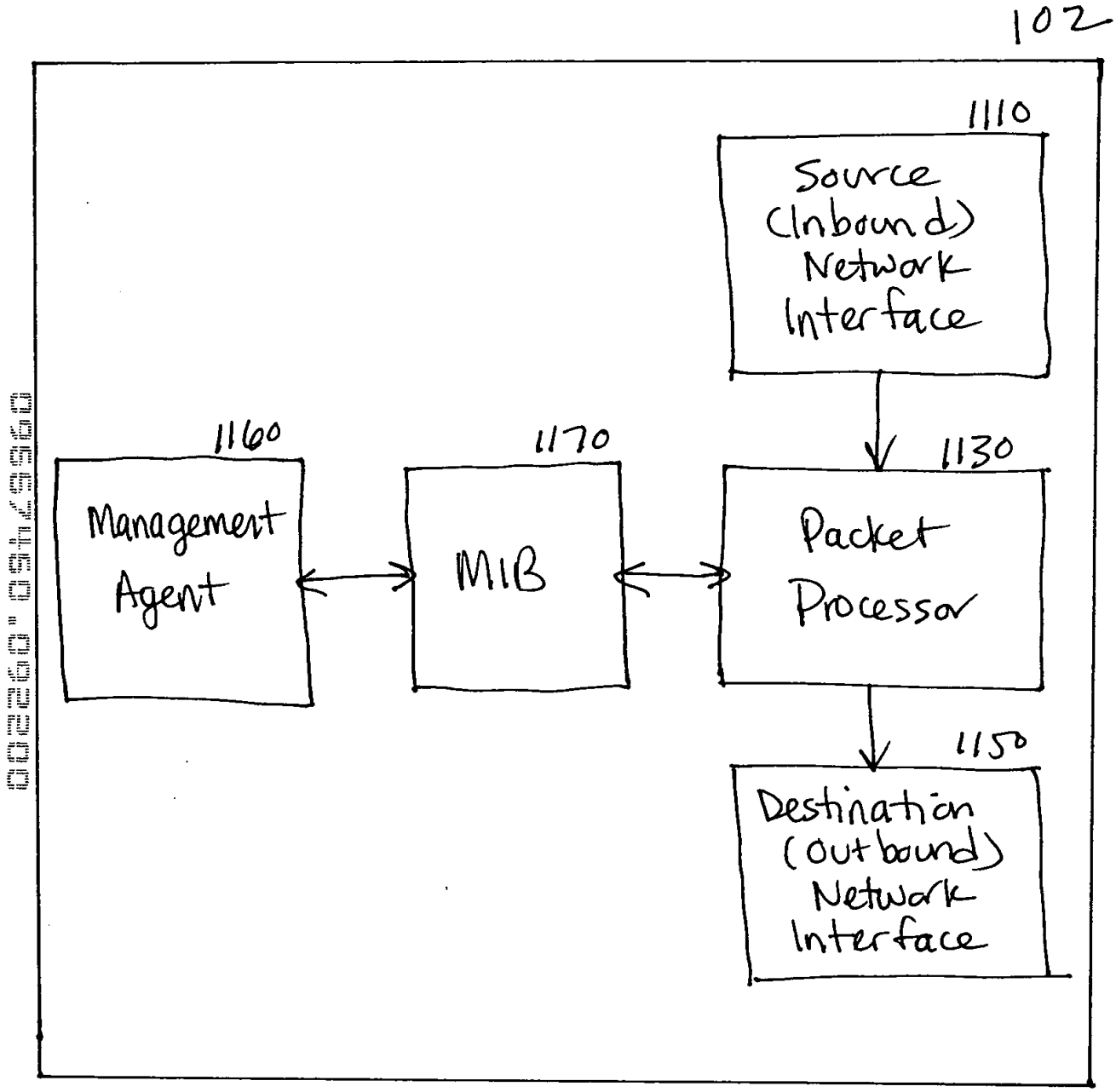
102
1110
Source (Inbound) Network Interface
1160
Management Agent
1170
MIB
1130
Packet Processor
1150
Destination (Outbound) Network Interface

FIG. 11 A

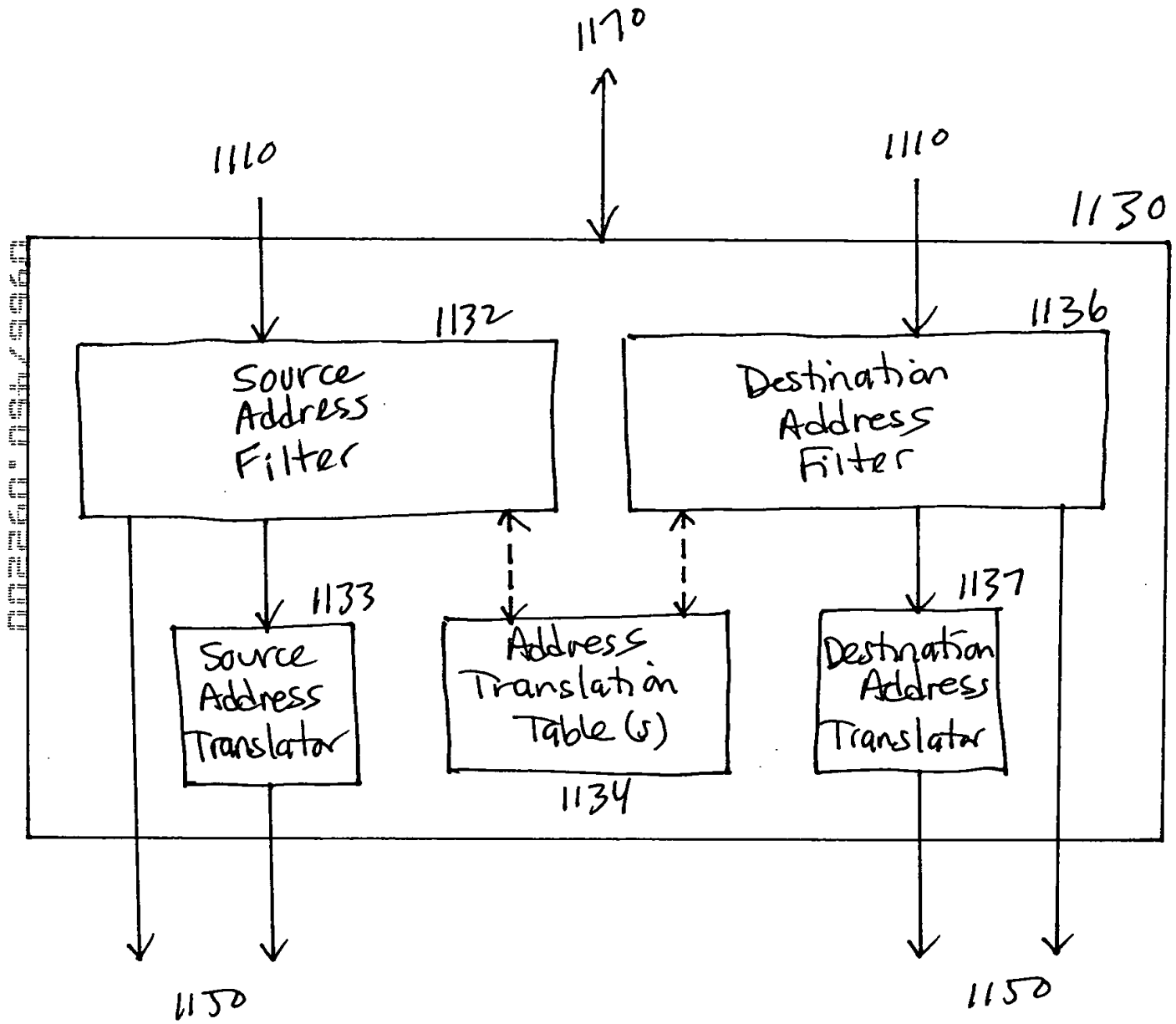
1170
1110
1110
1130
1132
Source Address Filter
1136
Destination Address Filter
1133
Source Address Translator
Address Translation Table(s)
1134
1137
Destination Address Translator
1150
1150

FIG. 11 B

```
Wellfleet-NAT-MIB DEFINITIONS ::= BEGIN

--

  IMPORTS

    IpAddress, Counter, Gauge
      FROM RFC1155-SMI
    OBJECT-TYPE
      FROM RFC-1212
    DisplayString
      FROM RFC1213-MIB
    wfNatGroup
      FROM Wellfleet-COMMON-MIB;

  wfNatIfTable OBJECT-TYPE
      SYNTAX   SEQUENCE OF WfNatIfEntry
      ACCESS   not-accessible
      STATUS   mandatory
      DESCRIPTION
            "The set of interface that are participating
            in the NAT protocol."
      ::= { wfNatGroup 6 }

  wfNatIfEntry OBJECT-TYPE
      SYNTAX   WfNatIfEntry
      ACCESS   not-accessible
      STATUS   mandatory
      DESCRIPTION
            "An single instance of a NAT interface entry."
      INDEX      { wfNatIfIpAddress,
           wfNatIfCircuit }
      ::= { wfNatIfTable 1 }
```

FIG. 12A

```
WfNatIfEntry ::= SEQUENCE {
    wfNatIfDelete
      INTEGER,
    wfNatIfDisable
      INTEGER,
    wfNatIfIpAddress
      IpAddress,
    wfNatIfCircuit
      INTEGER,
    wfNatIfType
       INTEGER,
    wfNatIfState
       INTEGER,
    wfNatIfTxCount
       Counter,
    wfNatIfRxCount
       Counter,
    wfNatIfPktDropCount
       Counter,
    wfNatIfDomain
      DisplayString
  }

wfNatIfDelete OBJECT-TYPE
  SYNTAX   INTEGER {
        created(1),
        deleted(2)
      }
  ACCESS   read-write
  STATUS   mandatory
  DESCRIPTION
        "This variable determines in a NAT Interface has been
        configured on the router."
  DEFVAL   { created }
  ::= { wfNatIfEntry 1 }

wfNatIfDisable OBJECT-TYPE
  SYNTAX   INTEGER {
        enabled(1),
        disabled(2)
      }
  ACCESS   read-write
  STATUS   mandatory
  DESCRIPTION
        "'The NAT interface's administrative status.  The value
        'enabled' denotes that NAT has been configured
        on the interface. The value 'disabled' denotes that
        the interface is not running NAT.'"
  DEFVAL   { enabled }
  ::= { wfNatIfEntry 2 }
```

FIG. 12B

```
wfNatIfIpAddress OBJECT-TYPE
   SYNTAX   IpAddress
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
        "The IP address of this NAT interface."
   ::= { wfNatIfEntry 3 }

wfNatIfCircuit OBJECT-TYPE
   SYNTAX   INTEGER
   ACCESS   read-write
   STATUS   mandatory
   DESCRIPTION
        "The circuit number of this interface."
   ::= { wfNatIfEntry 4 }

wfNatIfType OBJECT-TYPE
   SYNTAX   INTEGER {
              uniDirInbound(1),
              uniDirOutbound(2),
              biDirectional(3)
            }
   ACCESS   read-write
   STATUS   mandatory
   DESCRIPTION
        "Denotes  the type of NAT interface being defined."
   DEFVAL{ uniDirInbound }
   ::= { wfNatIfEntry 5 }

wfNatIfState OBJECT-TYPE
   SYNTAX   INTEGER {
            up(1),
            down(2),
            init(3)
         }
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
        "'The state of NAT on this interface'"
   DEFVAL   { down }
   ::= { wfNatIfEntry 6 }

wfNatIfTxCount OBJECT-TYPE
   SYNTAX   Counter
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
        "Number of packets mapped over this interface from
        the local to the global network.'"
   ::= { wfNatIfEntry 7 }
```

FIG. 12C

```
wfNatIfRxCount OBJECT-TYPE
   SYNTAX   Counter
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
         "Number of packets mapped over this interface from
         the global to the local network."
   ::= { wfNatIfEntry 8 }

wfNatIfPktDropCount OBJECT-TYPE
   SYNTAX   Counter
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
         "Number of packets dropped on this interface"
   ::= { wfNatIfEntry 9 }

wfNatIfDomain OBJECT-TYPE
   SYNTAX  DisplayString
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION
         "When wfNatIfType is set to biDirectional, specifies
        the Address Domain Name that this interface is
          connected to, otherwise set to null."
   ::= { wfNatIfEntry 10 }
```

FIG. 12 D

```
wfNatAddressRangeTable OBJECT-TYPE
   SYNTAX  SEQUENCE OF WfNatAddressRangeEntry
   ACCESS  not-accessible
   STATUS  mandatory
   DESCRIPTION
         "Table of address ranges. "
    ::= { wfNatGroup 8 }

 wfNatAddressRangeEntry OBJECT-TYPE
   SYNTAX  WfNatAddressRangeEntry
   ACCESS  not-accessible
   STATUS  mandatory
   DESCRIPTION
         "Information describing each of the available address ranges."
   INDEX   { wfNatAddressRangeAddress,
             wfNatAddressRangePrefixLen,
             wfNatAddressRangeIndex}
   ::= { wfNatAddressRangeTable 1 }

 WfNatAddressRangeEntry ::= SEQUENCE {
      wfNatAddressRangeDelete
         INTEGER,
      wfNatAddressRangeDisable
         INTEGER,
      wfNatAddressRangeAddress
         IpAddress,
      wfNatAddressRangePrefixLen
         INTEGER,
      wfNatAddressRangeIndex
         INTEGER,
      wfNatAddressRangeNto1Addr
         IpAddress,
      wfNatAddressRangeType
         INTEGER,
      wfNatAddressRangeDomain
        DisplayString,
      wfNatAddressRangeTransPool
        INTEGER,
      wfNatAddressRangeStaticNextHop
        IpAddress,
      wfNatAddressRangeUnnumCct
        INTEGER
   }
```

FIG. 12 E

```
wfNatAddressRangeDelete OBJECT-TYPE
   SYNTAX  INTEGER {
         created(1),
         deleted(2)
      }
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION
         "Create/Delete parameter. Default is created. Users perform
               a set operation on this object in order to create/delete
               an address range entry."
   DEFVAL  { created }
   ::= { wfNatAddressRangeEntry 1 }

wfNatAddressRangeDisable OBJECT-TYPE
   SYNTAX  INTEGER {
         enabled(1),
         disabled(2)
      }
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION
         "Enable/Disable parameter. Default is enabled. Users perform
               a set operation on this object in order to enable/disable
               an address range entry."
   DEFVAL  { enabled }
   ::= { wfNatAddressRangeEntry 2 }

wfNatAddressRangeAddress OBJECT-TYPE
   SYNTAX  IpAddress
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
         "The IP beginning address of this range entry."
   ::= { wfNatAddressRangeEntry 3 }

wfNatAddressRangePrefixLen OBJECT-TYPE
   SYNTAX  INTEGER (1 .. 32)
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
         "The number of contiguous bits set in the IP address mask
               which are used to define the address range of the entry."
   ::= { wfNatAddressRangeEntry 4 }

wfNatAddressRangeIndex OBJECT-TYPE
   SYNTAX  INTEGER
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
         "A unique value for this entry in wfNatAddressRangeTable."
   ::= { wfNatAddressRangeEntry 5 }
```

FIG. 12F

```
wfNatAddressRangeNto1Addr OBJECT-TYPE
   SYNTAX  IpAddress
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION
         "When wfNatAddressRangeType is set to srcAddrFilter,
                specifies the N-to-1 translation address used for
          this range, otherwise set to zero."
   DEFVAL  { 0 }
   ::= { wfNatAddressRangeEntry 6 }

wfNatAddressRangeType OBJECT-TYPE
   SYNTAX  INTEGER {
            sourceAddrFilter(1),
            translationPool(2),
            domainSrcAddrFilter(3),
            domainTransPool(4)
           }
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION
         "Denotes the type of address range being defined.
       sourceAddrFilter:       a range of IP addresses used to detect packets
                which need traditional NAT forwarding.
       translationPool:        for traditional NAT forwarding, a range of
                IP addresses from which translation addresses are picked.
       domainSrcAddrFilter:    a range of IP addresses used to detect domain
                specific packets which need domain specific NAT forwarding.
       domainTransPool:        for domain specific NAT forwarding, a range of
                IP addresses from which domain specific translation addresses
                are picked."
   DEFVAL{ sourceAddrFilter }
   ::= { wfNatAddressRangeEntry 7 }

wfNatAddressRangeDomain OBJECT-TYPE
   SYNTAX  DisplayString
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION
         "When wfNatAddressRangeType is set to  domainSrcAddrFilter or
                domainTransPool, specifies the Address Domain Name that this
                address range is valid for, otherwise set to null."
   ::= { wfNatAddressRangeEntry 8 }
```

FIG. 12G

```
wfNatAddressRangeTransPool OBJECT-TYPE
   SYNTAX  INTEGER {
           inbound(1),
           outbound(2)
         }
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION
        "This attribute is only valid for the wfNatAddressRangeType as
              domainSrcAddrFilter(3). The value of this attribute decides
              where to get the translation address for this range from.
              This could be either the translation Pool defined for the
              inbound domain or the outbound domain for the packet in
              question."
   DEFVAL{ outbound }
   ::= { wfNatAddressRangeEntry 9 }

wfNatAddressRangeStaticNextHop OBJECT-TYPE
   SYNTAX  IpAddress
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION
        "The IP address of the next hop of this range entry."
   DEFVAL{ 0 }
   ::= { wfNatAddressRangeEntry 10 }

wfNatAddressRangeUnnumCct OBJECT-TYPE
   SYNTAX  INTEGER
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION
        "This Nat Address range over the unnumbered interface."
   DEFVAL{ 0 }
   ::= { wfNatAddressRangeEntry 11 }
```

FIG. 12H

```
wfNatStaticMappingTable OBJECT-TYPE
   SYNTAX  SEQUENCE OF WfNatStaticMappingEntry
   ACCESS  not-accessible
   STATUS  mandatory
   DESCRIPTION
        "'This table creates instances of pre-defined NAT translations.'"
   ::= { wfNatGroup 9 }

wfNatStaticMappingEntry OBJECT-TYPE
   SYNTAX  WfNatStaticMappingEntry
   ACCESS  not-accessible
   STATUS  mandatory
   DESCRIPTION
        "A single original source address to translated address
              translation."
   INDEX   { wfNatStaticMappingTransAddress,
         wfNatStaticMappingProtocol,
         wfNatStaticMappingTransPort }
   ::= { wfNatStaticMappingTable 1 }

WfNatStaticMappingEntry ::= SEQUENCE {
      wfNatStaticMappingDelete
         INTEGER,
      wfNatStaticMappingDisable
         INTEGER,
      wfNatStaticMappingOrigAddress
         IpAddress,
      wfNatStaticMappingTransAddress
         IpAddress,
      wfNatStaticMappingProtocol
         INTEGER,
      wfNatStaticMappingOrigPort
         INTEGER,
      wfNatStaticMappingTransPort
         INTEGER,
      wfNatStaticMappingInDomain
            DisplayString,
      wfNatStaticMappingOutDomain
            DisplayString,
      wfNatStaticMappingStaticNextHop
         IpAddress,
      wfNatStaticMappingUnnumCct
         INTEGER
   }
```

FIG. 12 I

```
wfNatStaticMappingDelete OBJECT-TYPE
  SYNTAX  INTEGER {
        created(1),
        deleted(2)
      }
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
        "Create/Delete parameter. Default is created. Users perform a set
              operation on this object in order to create/delete a static
              address translation entry."
  DEFVAL  { created }
  ::= { wfNatStaticMappingEntry 1 }

wfNatStaticMappingDisable OBJECT-TYPE
  SYNTAX  INTEGER {
        enabled(1),
        disabled(2)
      }
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
        "Enable/Disable parameter. Default is enabled. Users perform a
              set operation on this object in order to enable/disable a static
              address translation entry."
  DEFVAL  { enabled }
  ::= { wfNatStaticMappingEntry 2 }

wfNatStaticMappingOrigAddress OBJECT-TYPE
  SYNTAX  IpAddress
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
        "The original (un-translated) address of the translation."
  ::= { wfNatStaticMappingEntry 3 }

wfNatStaticMappingTransAddress OBJECT-TYPE
  SYNTAX  IpAddress
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The translated address of the translation."
  ::= { wfNatStaticMappingEntry 4 }

wfNatStaticMappingProtocol OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
        "The IP protocol of the translation. Example values are
              6 for TCP, and 17 for UDP."
  ::= { wfNatStaticMappingEntry 5 }
```

FIG. 12J

```
wfNatStaticMappingOrigPort OBJECT-TYPE
   SYNTAX  INTEGER
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION
         "The original (domain specific) UDP or TCP port of the
               translation. This will only be relevent if the protocol
          is either UDP or TCP."
   ::= { wfNatStaticMappingEntry 6 }

wfNatStaticMappingTransPort OBJECT-TYPE
   SYNTAX  INTEGER
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
         "The translated UDP or TCP port of the translation. This will
               only be relevent if the protocol is either UDP or TCP."
   ::= { wfNatStaticMappingEntry 7 }

wfNatStaticMappingInDomain OBJECT-TYPE
   SYNTAX  DisplayString
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION
               "This attribute specifies the name of the address domain that this
               source translation shall be valid for. In other words, this
               translation shall only be valid for source addresses coming
               inbound from this domain."
      DEFVAL  { "private" }
   ::= { wfNatStaticMappingEntry 8 }

wfNatStaticMappingOutDomain OBJECT-TYPE
   SYNTAX  DisplayString
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION
               "This attribute specifies the name of the outbound address
               domain that this translation will be valid for. In other words,
               this translation only applies to translations that will be
               forwarded out into this address domain."
      DEFVAL  { "public" }
   ::= { wfNatStaticMappingEntry 9}

wfNatStaticMappingStaticNextHop OBJECT-TYPE
   SYNTAX  IpAddress
   ACCESS  read-write
   STATUS  mandatory
   DESCRIPTION
         "The IP address of the next hop of this static entry."
   DEFVAL{ 0 }
   ::= { wfNatStaticMappingEntry 10 }
```

FIG. 12K

```
wfNatStaticMappingUnnumCct OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "This Nat static translation over the unnumbered interface."
    DEFVAL{ 0 }
    ::= { wfNatStaticMappingEntry 11 }
```

FIG. 12 L

```
wfNatMappingTable OBJECT-TYPE
   SYNTAX  SEQUENCE OF WfNatMappingEntry
   ACCESS  not-accessible
   STATUS  mandatory
   DESCRIPTION
         "This table defines the current set of address translations
               that are in effect."
   ::= { wfNatGroup 10 }

wfNatMappingEntry OBJECT-TYPE
   SYNTAX  WfNatMappingEntry
   ACCESS  not-accessible
   STATUS  mandatory
   DESCRIPTION
         "A single original source address to translated address
               translation."
   INDEX   { wfNatMappingTransAddress,
             wfNatMappingProtocol,
             wfNatMappingTransPort }
   ::= { wfNatMappingTable 1 }

WfNatMappingEntry ::= SEQUENCE {
      wfNatMappingOrigAddress
         IpAddress,
      wfNatMappingTransAddress
         IpAddress,
      wfNatMappingProtocol
         INTEGER,
      wfNatMappingOrigPort
         INTEGER,
      wfNatMappingTransPort
         INTEGER,
      wfNatMappingTxCount
         Counter,
      wfNatMappingRxCount
         Counter,
      wfNatMappingTimeout
         Counter,
      wfNatMappingMode
            INTEGER,
      wfNatMappingInDomain
            DisplayString,
      wfNatMappingOutDomain
            DisplayString
       }

wfNatMappingOrigAddress OBJECT-TYPE
   SYNTAX  IpAddress
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
         "The original (un-translated) address of the translation."
   ::= { wfNatMappingEntry 1 }
```

FIG. 12M

```
wfNatMappingTransAddress OBJECT-TYPE
   SYNTAX  IpAddress
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
         "The translated address of the translation."
   ::= { wfNatMappingEntry 2 }

wfNatMappingProtocol OBJECT-TYPE
   SYNTAX  INTEGER
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
         "'The IP protocol of the translation.'"
   ::= { wfNatMappingEntry 3 }

wfNatMappingOrigPort OBJECT-TYPE
   SYNTAX  INTEGER
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
         "When the translation is for a TCP connection, this denotes
               the original TCP port number."
   ::= { wfNatMappingEntry 4 }

wfNatMappingTransPort OBJECT-TYPE
   SYNTAX  INTEGER
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
         "When the translation is for a TCP connection, this denotes
               the translated TCP port number."
   ::= { wfNatMappingEntry 5 }

wfNatMappingTxCount OBJECT-TYPE
   SYNTAX  Counter
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
         "Number of packets forwarded by NAT using this translation."
   ::= { wfNatMappingEntry 6 }

wfNatMappingRxCount OBJECT-TYPE
   SYNTAX  Counter
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
         "Number of packets received by NAT using this translation."
   ::= { wfNatMappingEntry 7 }
```

FIG-12 N

```
wfNatMappingTimeout OBJECT-TYPE
   SYNTAX  Counter
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
         "The time in seconds since this translation entry was last used.
               This is used to age out translation entries."
   ::= { wfNatMappingEntry 8 }

wfNatMappingMode OBJECT-TYPE
   SYNTAX  INTEGER
   ACCESS  read-only
   STATUS  mandatory
   DESCRIPTION
         "This is the bit mask representing the type of this translation.
               Each bit specifies the type as follows:

          The translation could be only ONE of the following three...

          0x01000000 - This translation is origianted on this router,
                                  i.e. this NAT router performed the translation.
          0x02000000 - This translation is learned from the peer,
                                  i.e. this translation was learned from the
                     peer using NAT Synchronization feature.
          0x04000000 - This translation is owned,
                                  i.e. it was originally  learned from peer, but
                                  this router received traffic which used this
                                  translation.

          ...and only ONE of the following three.

          0x00000010 - This translation is the STATIC translation.
          0x00000020 - This translation is Dynamic(1 to 1) translation.
          0x00000040 - This translation is N to 1 translation."
   DEFVAL  { 0 }
   ::= { wfNatMappingEntry 9 }

 wfNatMappingInDomain OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
                "This attribute specifies the name of the address domain that this
                 source translation shall be valid for. In ther words, this
                 translation shall only be valid for source addresses coming
                 inbound from this domain."
   ::= { wfNatMappingEntry 10 }
```

FIG. 12 O

```
wfNatMappingOutDomain OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
            "This attribute specifies the name of the outbound address domain
             that this translation will be valid for. In other words,  this
             translation only applies to translations that will be forwarded
             out into this address domain."
    ::= { wfNatMappingEntry 11}


END  -- Wellfleet-NPT-MIB
```

FIG. 12 P